

Memoria em Sistemas Operacionais Modernos

Rafael Figueira Goncalves
Universidade Estadual do Norte do Parana
Bandeirantes
rafael@itapecerica.com.br

ABSTRACT

Esse trabalho traz em formato *Extended Abstract* um breve resumo sobre o gerenciamento de memoria primaria em Sistemas Operacionais modernos, com base no Livro *Modern Operating Systems* de Andrew S. Tanenbaum [2]

Keywords

Memoria; Sistema Operacional

1. INTRODUÇÃO

A memoria é um componente essencial em um sistema moderno, conectado em nível de hardware a CPU e aos dispositivos de entrada e saída por meio do barramento de sistema. A cada ciclo, a CPU busca em memoria uma instrução para ser decodificada e então executada, por causa do alto custo computacional para acesso, CPUs modernos possuem registradores com função de buffer onde são mantidas instruções de frequente e recente acesso. Varias características da memoria tem um grande impacto potencial na performance geral do sistema, assim merecendo sua devida atenção.

2. MEMORIA

A memoria é classificada como segundo componente mais importante de um sistema (apos a CPU), diferentes tipos de tecnologias de memorias vieram evoluindo com o passar do tempo, porem no momento esta atrás dos recentes avanços de tecnologia implementada nos processadores modernos, a vasta maioria de tecnologias de memorias não são rápidas suficiente para fazer com que seja mais veloz que um ciclo de execução do processador, tendo assim uma situação ideal onde o processador não fica esperando, e sim processando a maior proporção de tempo possível, porem as poucas tecnologias capazes de fazer isso são financeiramente caras demais para ser implementada exclusivamente por todo o sistema, por isso são usadas camadas de tipos diferentes em uma hierarquia, que serão exploradas a seguir.

2.1 Registradores

Registradores estão na primeira camada de memoria, são integrados a CPU e por isso seu tempo de acesso é o mais curto possível, se custo financeiro e espaço físico não fossem levados em consideração, toda a memoria primaria seria em registradores, porem no estado atual o usual é menos de 1 KB de registradores em todo o sistema, esse espaço deve ser gerenciado via software, decidindo o que manter ou nao em registradores.

2.2 Memoria Cache

Memoria Cache, normalmente controlada via hardware, seu nivel mais rapido se encontra dentro ou o mais perto possivel da CPU. Consideravelmente maior que os registradores mas com uma velocidade ainda rapida (relativa mas nao acima da CPU) serve de meio termo entre a memoria principal e os registradores, quando um programa precisa acessar algo em memoria, o hardware de cache checa se existe em cache, poupando o custo computacional de acessar a memoria principal. Existem múltiplos níveis de memoria cache, nível 1 ou L1 cache esta sempre dentro do pacote da CPU, normalmente entrega instruções já decodificadas diretamente a CPU para execução. L2 cache normalmente bem maior que L1 (múltiplos megabytes comparados a poucos kilobytes) é usado para guardar palavras de memoria recentemente usados. Enquanto L1 tem um acesso imediato, L2 tem um atraso de 1 a 2 ciclos de clock. Em chips com múltiplos núcleos, existem varias opções de implementação do cache, como compartilhamento do cache entre 2 ou mais núcleos, o que requer um controlador mais complexo porem traz uma sincronia mais fácil comparado a implementação individual.

2.3 Memoria Principal

Memoria Principal também conhecida como RAM (Memoria de Acesso Randômico) é onde se mantém tudo aquilo que não cabe na memoria cache. Como os tipos de memoria visto previamente, RAM é volátil, isto é, perde o seu conteúdo uma vez que o sistema não esteja energizado.

2.4 ROM

Muitos computadores contem ROM (Memoria de Somente Leitura), não volátil, programado em fabrica e não podendo ser alterado pelo usuário final.

2.4.1 EEPROM

EEPROM é um tipo de ROM que pode ser apagada e escrita, porem com grande custo em tempo, assim não sendo útil em aplicações como de RAM, uso ideal consiste do mesmo

que ROM porem com a vantagem de poder ser escrito caso haja necessidade como um update ou remoção de erros.

2.4.2 Memoria Flash

Flash é outro tipo de memoria não volátil, utilizada em dispositivos eletrônicos portáteis, intermediário em velocidade entre RAM e disco, porem com uma vida útil de ciclos de escrita menor.

2.5 Disco Mecânico

Na ultima camada da hierarquia achamos o disco magnético, duas ordens de magnitude mais barato e maior que RAM, porem seu acesso, principalmente o acesso randômico é três ordens de magnitude mais lento pois deve se movimentar fisicamente a ponta da agulha para a posição correta do disco, comparado com RAM ou flash cujo acesso é puramente eletrônico e como resultado muito mais rápido

2.6 SSDs

Discos de Estado Solido são mais semelhantes a RAM que a discos mecânicos, pois não tem partes que se movem, seus dados são armazenados em flash, um meio termo em performance, capacidade e custo entre RAM e discos mecânicos

3. GERENCIAMENTO DE MEMORIA

Memoria é um recurso que deve ser cuidadosamente e devidamente gerenciada, cabe ao Gerenciador de Memoria, uma parte do sistema operacional decidir como.

3.1 Abstrações

Os primeiros computadores gerenciavam memoria sem algum tipo de abstração, cada programa acessava a memoria física, não sendo possível a execução de dois programas em paralelo, pois as alteracoes de um corromperia os dados do outro. Soluces para execução "paralela" de múltiplos programas envolve o armazenamento dos dados em memoria de um programa ao disco, e carregamento do próximo, somente um programa pode estar em memoria por vez. Outras solucoes via implementação de hardware customizado surgiram para solucionar esse problema, porem a pratica mais usual em sistemas modernos consiste do uso de abstrações de memoria.

3.1.1 Address Spaces

Resolvendo o problema de ter a memoria física exposta aos processos surgiu a abstração de pacos de endereços, resolvendo dois problemas para habilitar a execução de múltiplas aplicações em memoria ao mesmo tempo sem interferência entre elas, essa abstração cria um espaço onde cada programa fica. O espaço de endereços é composto de uma lista de endereces que um processo pode usar para acessar um espaço em memoria predestinado a aquele processo. A forma mais simples de implementar essa abstração é com registradores de base e limite, quando os dois são usados por uma aplicação, a próxima posição em memoria é alocada se houver espaço livre.

3.1.2 Swapping

Swapping é usado em sistemas modernos quando a memoria física não é capaz de suportar todos os programas requeiridos, quando isso acontece o programa menos utilizado sera transferido para o disco e retornado a RAM uma vez feita uma requisição da CPU, processos inativos são mantidos em disco

3.1.3 Memoria Virtual

Com os avanços em tamanho de programas, outra solução é necessária além do Swapping, visto que acaba se tornando um processo lento por depender da velocidade do disco utilizado no sistema, surgiu então a memoria virtual que consiste de cada programa sendo executado conter o seu próprio espaço de endereços segmentado em paginas, e as paginas são mapeadas na memoria física porem nem todas as paginas necessitam estar em memoria física ao mesmo tempo para que o programa seja executado com sucesso. Quando o programa faz uma requisição de um espaço de endereço que esta em memoria física, o hardware simplesmente executa o mapeamento necessário, quando uma parte do espaço de endereço não se encontra na memoria física, sintema operacional acessa a peca que falta e re-executa a instrução que falhou

3.1.4 Paging

Paging é uma técnica utilizada por vários sistemas de memoria virtual por via de endereços virtuais, um caminho relativo a memoria física, gerada por software. Endereços são interceptados pela MMU (Unidade de Gerenciamento de Memoria) que remapeia o endereço virtual na memoria física

4. CONCLUSÕES

Enquanto não houver a memoria perfeita citada por Tanenbaum, que seja privada, infinitamente grande, infinitamente rápida, não volátil e econômica, gerenciamento de memoria continuara sendo um elemento fundamental para a performance geral de um sistema.

5. REFERENCES

- [1] A. Iqbal. *Operating Systems MCQs: Multiple Choice Questions and Answers*. 2016.
- [2] A. S. Tanenbaum. *Modern operating systems*. Pearson Education, 2009.

APPENDIX

A. CRITERIO DE SELELECAO DO TEMA DESTE ARTIGO

Todas questoes de multipla escolha retiradas do seguinte livro: Operating Systems MCQs: Multiple Choice Questions and Answers (Quiz and Tests with Answer Keys) [1] de Arshad Iqbal (acompanhado de seu website [?]) Distribuiçao aproximada dos subtemas por questao:

* Memory 2, 5, 6, 11, 19
Process 16, 17, 18, 20
Interrupts 3 4, 7, 15
I/O 1, 13, 14
Multi 9, 12
Microkernel 10
Scheduler 8

A.1 Fontes

Livros referenciados para justificacoes, de Andrew S Tanenbaum:
Structured Computer Organization (6th edition) [1]
Modern Operating Systems (4th edition) [2]

A.2 Questões

1. I/O function allows to exchange data directly between an

- a. Process States
- b. Registers
- c. I/O module and the processor
- d. I/O devices

Correta = c, como no primeiro paragrafo de 3.7 "Interfacing" na pagina 232 [1], I/O tem a funcao de fazer o computador se comunicar com o mundo externo

Errada = a, nao foi citado em 5.8.4 na pagina 424 [1] o uso de I/O para alterar estados de processos

Errada = b, como deduzido do primeiro paragrafo de 5.1.3 na pagina 349 [1], os registradores nao se relacionam com as funcoes de I/O

Errada = d, como descrito em todo o capitulo "Interfacing" iniciado na pagina 349 [1], formas de comunicacao somente entre dispositivos e computadores por meio do I/O, nao entre dispositivos distintos.

2. Cache memory is intended to provide memory access

- a. Fastest
- b. Slow
- c. Very Slow
- d. Fast

Correta = a, em 4.5.1 na pagina 304 [1] foi descrito a necessidade que proporcionou o desenvolvimento da memoria cache, em que as memorias RAM continuam sendo melhoradas porem a um nivel muito abaixo das velocidades de processamentos das CPUs, assim atrapalhando o rendimento total, a solucao foi implementar uma memoria temporaria entre a RAM e os registradores, com baixa latency e velocidade extremamente alta, muitas vezes em paralelo para aumentar ainda mais os seus beneficios.

Incorretas = b,c,d, visto que o cache foi projetado para ser o mais rapido possivel

3. Interrupts are provided primarily as a way to

- a. Improve processor utilization
- b. Improve processor execution
- c. Improve processor control
- d. Improve processor speed

Correta = a, como descrito em 5.6.5 na pagina 414 [1], as interrupções alteram o fluxo de controle quando por exemplo um processo é finalizado, a CPU então acha um novo processo por via do vector de interrupção, diminuindo assim os ciclos onde o processador esta idle, aumentando a utilização. Incorretas = b,c e d, Interrupcao nao afeta essas outras variaveis

4. Program counter contains the address of the

- a. Next programs to be fetched
- b. Previous programs to be fetched
- c. Previous information to be fetched
- d. Next information to be fetched

Correta = d, como descrito no passo 6 de "Hardware Ac-

tions" na pagina 415 [1], o program counter aponta para a proxima informacao.

Incorreta = a, como descrito no mesmo passo, o program counter nao apontara necessariamente para uma aplicacao, no exemplo dado ele apontaria para uma rotina de servicos provenientes do dispositivo causando a interrupcao.

Incorretas = b e c, nao achei mencao em [1] de backtracking/backlogging necessario para retonar em processos anteriores.

5. Cache size issue can have significant impact on

- a. input
- b. output
- c. Information
- d. Performance

Correta = d, dito no quarto paragrafo da pagina 84 [1], performance esta positivamente relacionada ao tamanho do cache

Incorretas = a e b, como descrito no mesmo paragrafo, quanto maior o cache, cria-se o problema de maior o custo de acesso

Incorreta = c, tamanho do cache nao afeta a informacao nele contida

6. Data and instructions that are being used frequently are stored in

- a. Cache
- b. Block
- c. hard disk
- d. main memory

Correta = b, como descrito no terceiro paragrafo da pagina 84, as memorias sao subdevididas em blocos, no caso de cache

Incorreta = a, como descrito na pagina 83 [1], é usado para manter dados de rápido acesso

Incorreta = c, como dito na pagina 86, esse tipo de memoria é extremamente lenta (porem de maior capacidade), comparada com as de nível acima, não seria usado para data e instruções de acesso frequente

Incorreta = d, abstracao que inclui outros tipos de memoria como visto no capitulo 2

7. In implementation of Semaphores, for a single processor system, it is possible to inhibited

- a. Deadlock
- b. Interrupts
- c. Lock Step
- d. None

Correta = b, no segundo capitulo da pagina 480 [1] o autor cita o "truque" que funciona somente para sistemas com um unico processador, ao desabilitar os interrupts, assim o processador deve terminar um processo antes de comecar o outro

8. Time sharing technique handles

- a. Single Interactive Job

- b. Multiple Interactive Job
- c. Recent Interactive Job
- d. Old Interactive Job

Correta = b, como mencionado em "Introduction to Scheduling" na pagina 150 [2] timesharing é uma das técnicas utilizadas para gerenciar múltiplos programas

9. With the use of multiprogramming batch processing work can be

- a. Efficient
- b. Rigid
- c. Expensive
- d. Flexible

Correta = a, Descrito em "introduction to Scheduling" na pagina 150 [2] a multiprogramacao lida com o fato de escalonador dever tratar requisicoes de multiplos usuarios.

10. With Microkernel architecture it is possible to handle hardware interrupts as

- a. Application
- b. Information
- c. Data
- d. Message

Correta = d, em "Microkernels" quarto paragrafo da pagina 66, foi citado a possivel comunicacao entre processos utilizando interrupts

11. The owner of an address space can grant a number of its

- a. Modules
- b. Pages
- c. Devices
- d. Computers

Correta = b, em "Memory-Mapped I/O" na pagina 342 [2] o autor explica que com memoria mapeada nao ha necessidade de mecanismos de protecao externos para impedir que usuarios performem I/O, cada usuario tem acesso a sua tabela de paginas.

12. A common synchronization mechanism used in multiprocessor operating system is

- a. Complex
- b. Locks
- c. Lockstep
- d. None

Correta = b, em "Multiprocessor Synchronization" pagina 535 [2] o autor cita o uso da sincronizacao por bloqueio

13. I/O interrupt-driven is more efficient than

- a. I/O Modules
- b. I/O Devices
- c. Programmed I/O
- d. CPU

Correta = c, em 5.2.2 "Programmed I/O" na pagina 252 [2] o autor cita que que entrada e saída programada é a forma mais simples de I/O, onde a CPU faz todo o trabalho

14. I/O modules performs the requested action on

- a. Programmed I/O
- b. Direct Memory Access (DMA)
- c. Interrupt driven I/O
- d. I/O devices

Correta = a, em "Modules in Linux" na pagina 755 [2] o autor cita o funcionamento do Programmed I/O ao checar periodicamente ate o fim da operacao

15. Secure system concerned with protecting the system against

- a. User
- b. Security
- c. Interruption
- d. Damage

Correta = c, em 9.4 "Formal Models of Secures Systems" na pagina 611 [2] o autor explica como é usada a matriz de proteção para gerencial qual o próximo passo de um processo no domínio determinado

16. The concept of process for the structure of operating system is

- a. Fundamental
- b. Effective
- c. Old
- d. Modern

Correta = a, (e b) em 1.5.1 "Processes" na pagina 39 [2], o autor define processos como um "conceito chave", fundamentalmente necessario para conter toda informacao necessaria para execucao de um programa.

17. Thread is a dispatch able unit of

- a. Program
- b. Work
- c. Time
- d. Process

Correta = a, em 2.2.1 "Thread Usage" na pagina 97 [2] o autor explica a razao para a multiplicidade das threads e como sao facilmente destruidas pelos seus processos pais

18. In processes, access control implements a

- a. Security Policy
- b. Access Policy
- c. Control Policy
- d. Check Policy

Correta = a, em 9.3.2 na pagina 605 e 606 é ilustrada a lista de controle de acesso, supostamente com as funções das outras alternativas inclusas

19. Cache manager improves the performance of

- a. Programmed I/O
- b. File base I/O
- c. I/O device
- d. I/O Modules

Correta = b, em 11.6 "Caching in Windows" na pagina 942 [2] o autor confirma que o cache melhora a performance dos sistemas de arquivos, mantendo os arquivos de uso recente e frequente em memoria, em especifico em NTFS como ele armazena todos os dados como arquivos, incluindo a meta-data do file system

20. Process is a collection of

- a. Threads
- b. Files
- c. Registers
- d. Buffers

Correta = a, em 2.2 "Threads" na pagina 97 [2], o autor cita que em sistemas operacionais tradicionais cada processo tem somente uma thread, porem em muitas situações ter múltiplos threads é algo favorável

Incorreta = b, como descrito nas paginas 41 e 42 [2] arquivos sao tratados pelo file system

Incorreta = c, descrito em 3.3.3 "Registers" na pagina 174 [1] registradores sao itens de memoria